¿Cómo usar nuestro proyecto?

Acceder a nuestro repositorio:

Nuestro repositorio se encuentra alojado en Github en la siguiente dirección:

https://github.com/Luchit0-afk/Beca-VES-Selenium/tree/main

Selenium y Jenkins:

Para empezar aclaramos que los pasos a seguir para inicializar y ejecutar jenkins fueron hechos en Ubuntu 20.04. No notamos diferencia usando otros sistemas operativos pero no podemos asegurarnos de ello.

Inicializacion de Jenkins:

Para inicializar ***Jenkins*** se debe iniciar una terminal y seguir los siguientes pasos:

1. Ir al directorio donde descargo nuestro proyecto(que se puede descargar o clonar desde el siguiente enlace: git@github.com:Luchit0-afk/Beca-VES-Selenium.git)
2. Acceder al directorio **Jenkins**
3. Ejecutar el siguiente comando: *java -jar jenkins.war*
4. Usar un navegador para acceder a la siguiente ruta: **http://localhost:8080/**

Si se cuenta con el puerto 8080 ocupado en la PC recomendamos ejecutar jenkins en un puerto diferente usando el siguiente comando: *java -jar jenkins.war --httpPort=*número del puerto**. Para más información consultar la documentación oficial (https://www.jenkins.io/doc/book/installing/initial-settings/).

Importante: Para usar jenkins se debe contar con una versión de java 8 a 11(a dia de la fecha 26/10/21)

Configuración inicial de Jenkins:

Una vez corriendo Jenkins lo que debemos hacer es una configuración inicial. Apenas entramos a la ruta donde está ejecutándose Jenkins podemos ver que nos pide una contraseña y nos da una ruta para buscar esa contraseña. Evidentemente, debemos ir a la ruta que nos indica, copiar la contraseña y pegarla en el lugar correspondiente.(Recomendamos guardar esta contraseña en un lugar ajeno a la ruta por si en algún momento tenemos inconvenientes en un futuro.)

Una vez ingresada la contraseña, nos preguntará qué plugins instalamos para acompañar la aplicación. Recomendamos elegir la opción de instalar los plugins sugeridos ya que es la opción elegida por nosotros. Si en algún momento del proyecto debemos instalar otro plugins, lo anotaremos en esta misma documentación.

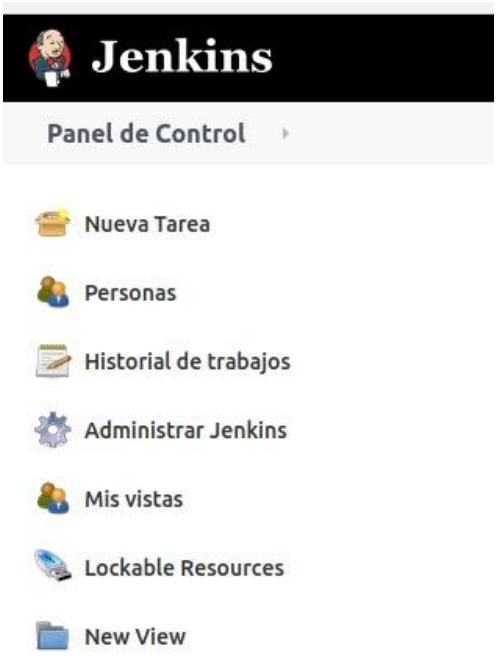
Por último y para finalizar la configuración inicial, Jenkins nos pedirá que creemos un usuario y contraseña. Recomendamos guardar estos datos ya que no podrán ser modificados en caso de olvidarnos de estos.

Una vez creado el usuario y contraseña, podremos ejecutar entrar a Jenkins. Ahora cada vez que ejecutemos jenkins nos pedirá estos datos para ingresar.

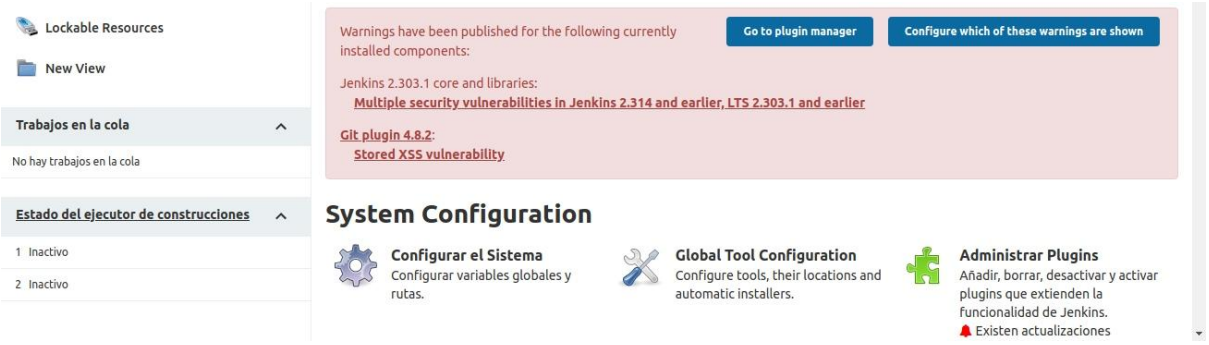
<u>Configurar la variable de entorno:</u>

Para ejecutar nuestro proyecto maven hecho en eclipse usando la herramienta Selenium debemos configurar dentro de Jenkins la variable de entorno de la siguiente manera:

1. Debemos ingresar en la sección de *Administrar Jenkins*:



2. Ahora dentro de nuestra seccion debemos ingresar a la parte de *Configurar el sistema:*



3. Ahora lo que debemos hacer es deslizar para abajo nuestra pestaña hasta encontrar la sección *Propiedades Globales.* Dentro de esta sección debemos configurar la variable de entorno de la siguiente manera(nombre: JAVA_HOME, valor: /usr/lib/jvm/java-11-openjdk-amd64/).

Importante: En la imagen figura la configuración para Ubuntu 20.04 y particular para Java 11. Si tiene Ubuntu 20.04 pero un java diferente recomendamos leer la siguiente documentación para configurar a java como variable de entorno en su PC: https://remot-technologies.com/como-anadir-java_home-a-las-variables-de-entorno-de-ubuntu-20-04-lts/ y poner en “valor” lo arrojado por el comando: *echo $JAVA_HOME*

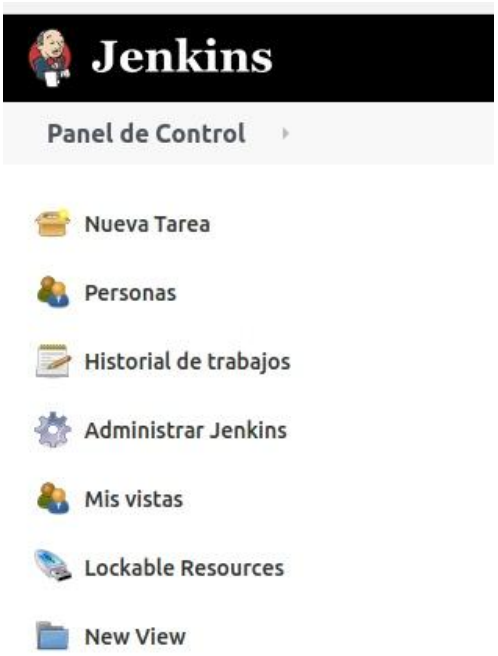
Si tiene otro sistema operativo recomendamos buscar un tutorial adecuado en internet que lo ayude en esta parte.

4. Por último debemos deslizarnos para abajo y aplicar y guardar los cambios.

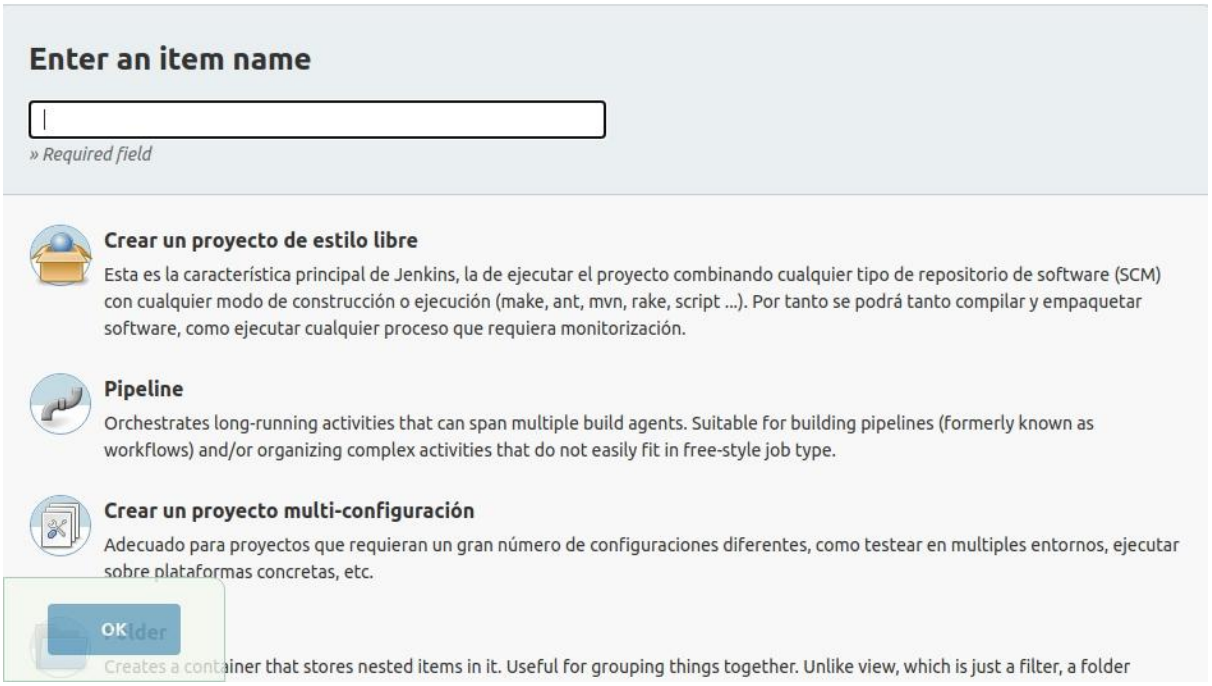
<u>Crear el proyecto:</u>

Una vez dentro de Jenkins lo que debemos hacer es crear un nuevo proyecto. Lo haremos siguiendo los siguientes pasos:

1. Debemos ingresar en la sección de *nueva tarea*:



2. Debemos ingresar el nombre que queramos que tenga la tarea , seleccionar donde dice “Crear un proyecto de estilo libre” y darle en *OK.*

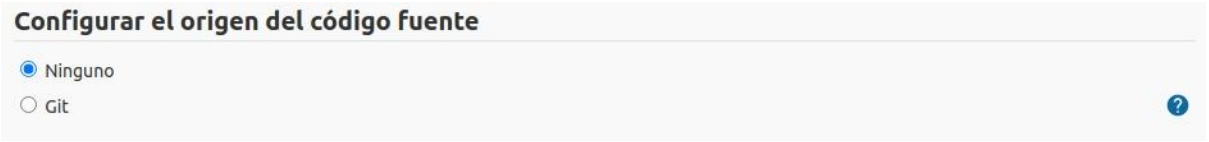


3. Por último debemos configurar nuestro proyecto deslizando para abajo para encontrar las siguientes secciones:

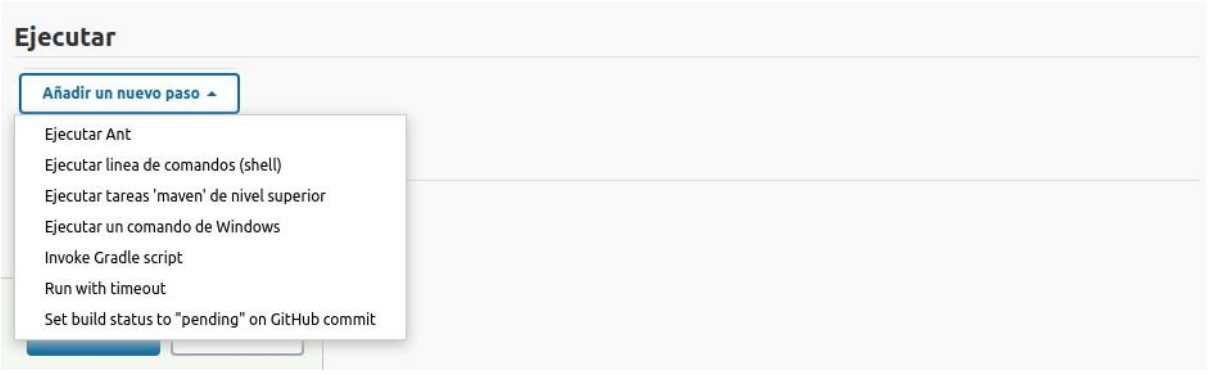
1. En la sección *JDK* debemos elegir la opcion de System:

Importante: Si esta opción no nos aparece el JDK vendrá definido por defecto y no debemos configurar nada en este caso.

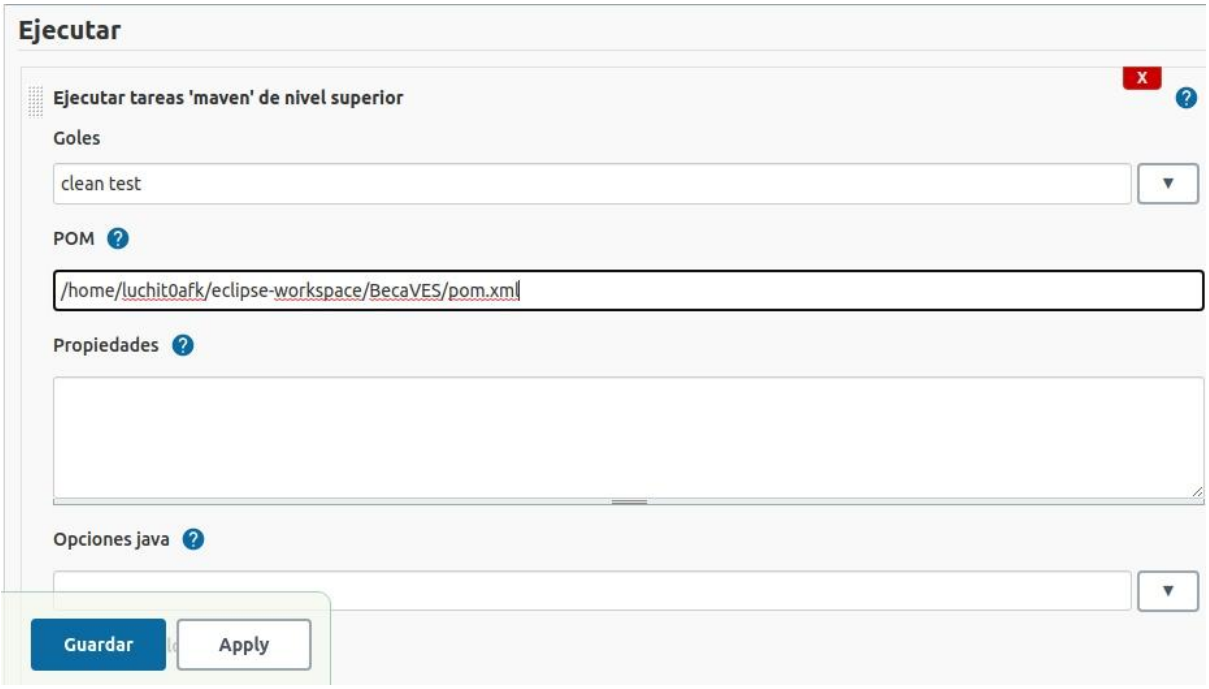
2. En la sección *Configurar el origen del código fuente* debemos elegir(por ahora) la opción de ninguno:



3. En la sección *Ejecutar* debemos desplegar la lista desplegable *Añadir un nuevo paso* y elegir la opción *Ejecutar tareas 'Maven' de nivel superior:*



Posteriormente, debemos elegir los siguientes valores:



Importante: Evidentemente en el valor *POM* debemos poner la ruta donde esta el pom de nuestro proyecto maven.

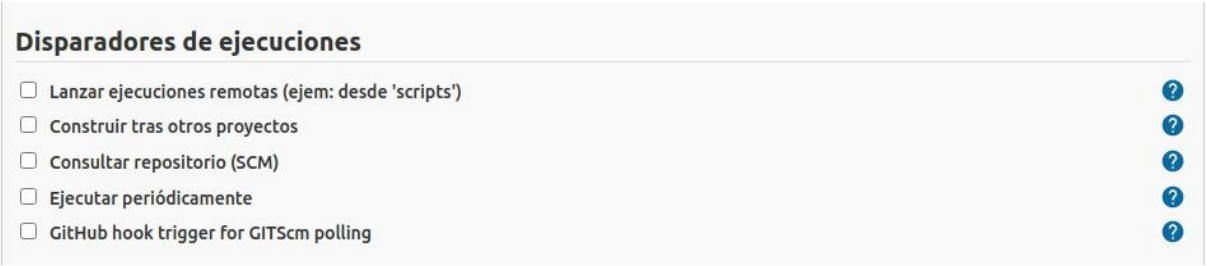
4. Finalmente, debemos guardar los cambios.

Ejecutar el proyecto periódicamente:

Tal vez en algún momento vamos a querer ejecutar el proyecto un día específico a una hora específica o ejecutar el proyecto todos los días a una cierta hora, entre otros casos parecidos. Para configurar esto podemos seguir los siguientes pasos:

1. Para empezar, debemos entrar a la configuración de nuestro proyecto. Esto lo hacemos accediendo a nuestra página principal de Jenkins, clickeando en nuestro proyecto que se identificara con el nombre puesto cuando lo creamos y accediendo a la sección *Configuración* que aparece a la izquierda de la pantalla.

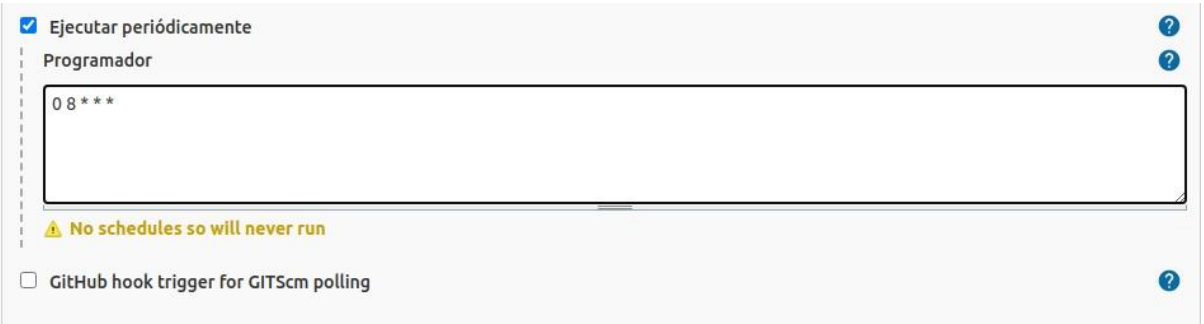
2. Deslizamos hasta llegar a la sección *Disparadores de ejecuciones* y clickeamos en la sección *Ejecutar periódicamente.*



3. Finalmente podemos establecer un horario de ejecución. Para esto nosotros debemos escribir 5 números en orden que representan lo siguiente:

1. MINUTOS: Minutos en una hora (0-59)
2. HORAS: Horas en un día (0-23)
3. DIA MES: Dia en un mes (1-31)
4. MES: Mes en un año (1-12)
5. DAYWEEK: Día de la semana (0-7) donde 0 y 7 son los domingos

Ejemplos de uso sería escribir el siguiente horario de ejecución:



Así establecemos que el proyecto se ejecutara todos los dias a las 8:00.

Así podríamos establecer que el proyecto se ejecute cada 5 minutos.

Por último, podemos usar un parámetro *h.* Este parámetro sirve para permitir que las tareas programadas periódicamente produzcan una carga uniforme en el sistema, recomendamos usar el símbolo h(para “hash”) siempre que sea posible.

Por ejemplo, usar 0 0 * * * para una docena de trabajos diarios causará un gran aumento a la medianoche. Por el contrario, el uso de H H * * * aún ejecutaría cada trabajo una vez al día, pero no todos al mismo tiempo, mejor utilizando recursos limitados.

Importante: El símbolo H se puede considerar como un valor aleatorio en un rango, pero en realidad es un hash del nombre del trabajo, no una función aleatoria, por lo que el valor permanece estable para cualquier proyecto dado.

Integrar Jenkins con un repositorio Github:

Integrar nuestro proyecto en Jenkins con su repositorio asociado nos será de mucha utilidad, ya que nos proveerá muchas funcionalidades nuevas. Entre ellas, podremos ejecutar nuestro proyecto jenkins cada vez que haya un commit a una branch específica de nuestro repositorio. Antes de entrar con las funcionalidades veamos cómo integrar jenkins con Github:

Pasos a seguir en Jenkins:

1. El primer paso es entrar a la configuración de nuestro proyecto(como hemos explicado anteriormente en la documentación) e irnos hasta la sección *Configurar el origen del código fuente.*
2. Una vez en esta sección clickeamos en la opción *Git* y apretamos el botón *Add Repository.* Aquí debemos poner el URL de nuestro repositorio como figura en la imagen a continuación:

3. Por último, debemos indicar la branch a partir de la cual se construirá el proyecto. Debemos clickear el boton *Avanzado* y en la sección *Branches to build* debemos indicar el nombre de nuestra branch(en mi caso la branch es main):

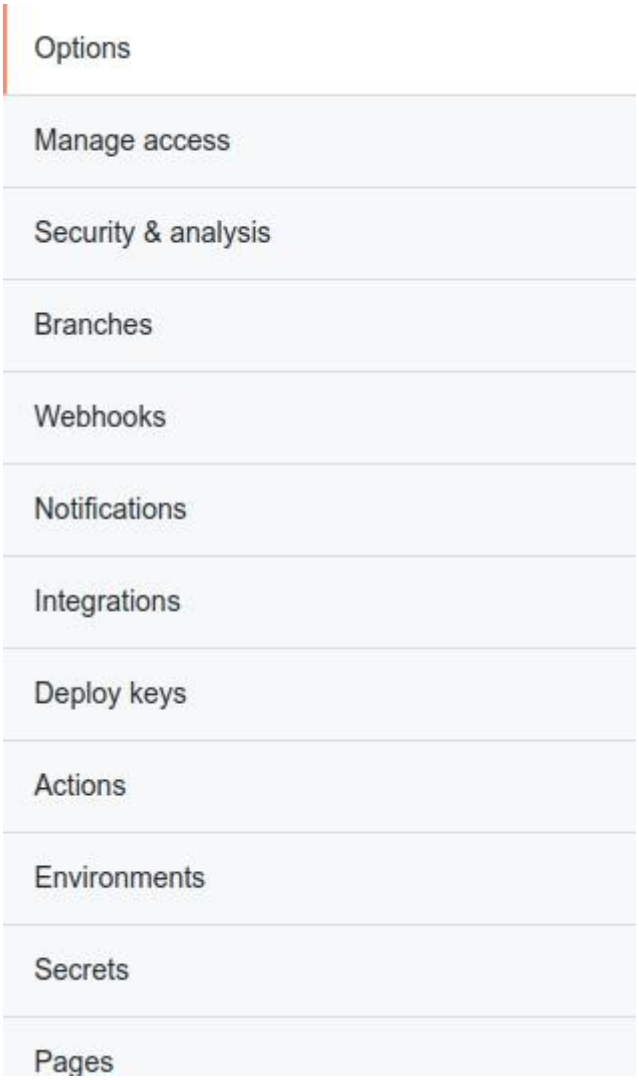
Importante: Por defecto esta branch es master pero si en nuestro repositorio no existe esta branch, el proyecto no se podra contruir ni ejecutar(Para mas informacion sobre esta configuracion visitar: https://www.cloudbooklet.com/jenkins-how-to-build-a-specific-branch-on-github/ , https://www.jenkins.io/doc/book/pipeline/multibranch/ o cualquier documentacion a fin)

4. Finalmente, debemos aplicar y guardar los cambios.

Pasos a seguir en GitHub:

1. Ingrese su nuestro repositorio y ingrese a la sección *Setting* en la parte superior:

2. Una vez dentro de la sección *Setting,* localice las pestañas de configuración(parte izquierda) e ingrese a la de Webhooks:



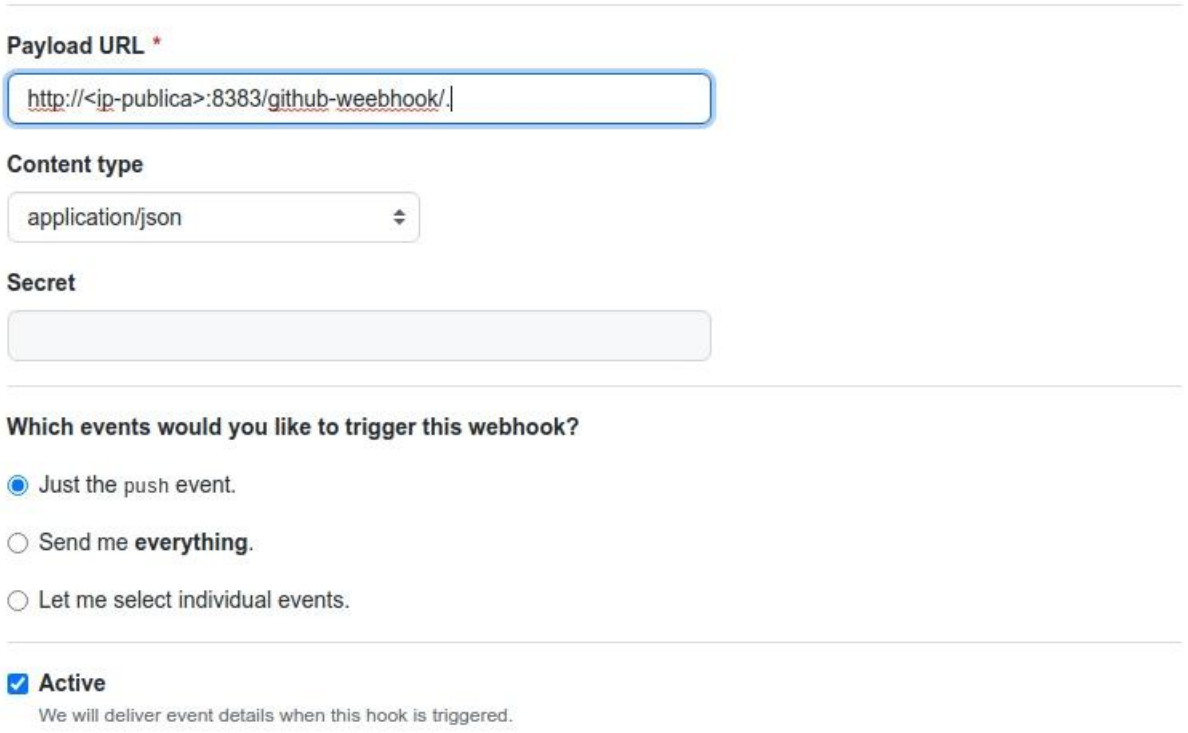
3. Una vez dentro añada una nueva Webhook clickeando el botón *Add webhook*:

4. Una vez dentro, rellenaremos los campos de la siguiente manera:
    a. En la sección *Payload URL* debemos ingresar el url(con ip pública) donde se está ejecutando Jenkins con el sufijo “/github-weebhook/”. En caso de que estemos corriendo

Jenkins en nuestra computadora deberemos averiguar cual que es nuestra ip publica en esta pagina. Por ejemplo, si estoy corriendo Jenkins en localhost y puerto 8383, lo que debemos ingresar aqui sería http://<ip-publica>:8383/github-weebhook/.

b. En la sección *Content type* debemos seleccionar la opción *application/json*.
c. En la seccion *Which events would you like to trigger this webhook?* Debemos seleccionar la opción *Just the push event.*
d. Por último, debe estar activado el tilde de la opción *Active.*

Un ejemplo de como quedaria es:



5. Finalmente clickeamos en *Add webhook.*

Ejecutar nuestro proyecto por cada commit a nuestro repositorio:

Una funcionalidad muy útil de Jenkins es que podemos configurar nuestro proyecto para que se ejecute cada vez que detecte un cambio en nuestro repositorio. Para hacer esto debemos tener integrado GitHub con Jenkins como hicimos en esta misma documentación.

Lo único que debemos hacer es entrar a la configuración de nuestro proyecto(como hemos explicado anteriormente en la documentación) e irnos hasta la sección *Disparadores de ejecuciones,* dentro de esta seleccionar la opción de *Consultar repositorio (SCM)* y dentro de esta ingresar “* * * * *”. Aplicamos, guardamos y listo. La configuración estaría completa.

Deberia quedar asi: